Create account Log in

Article Talk Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Navigation
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Data item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
Esperanto
Français
日本語
Русский
Српски / srpski
Edit links

# HTTP ETag

From Wikipedia, the free encyclopedia

The **ETag** or **entity tag** is part of HTTP, the protocol for the World Wide Web. It is one of several mechanisms that HTTP provides for web cache validation, and which allows a client to make conditional requests. This allows caches to be more efficient, and saves bandwidth, as a web server does not need to send a full response if the content has not changed. ETags can also be used for optimistic concurrency control,[1] as a way to help prevent simultaneous updates of a resource from overwriting each other.

**HTTP**
Persistence · Compression · HTTPS ·
**Request methods**
OPTIONS · GET · HEAD · POST · PUT · DELETE · TRACE · CONNECT · PATCH ·
**Header fields**
Cookie · **ETag** · Location · HTTP referer · DNT · X-Forwarded-For ·
**Status codes**
301 Moved Permanently · 302 Found · 303 See Other · 403 Forbidden · 404 Not Found ·
v · t · e ·

An ETag is an opaque identifier assigned by a web server to a specific version of a resource found at a URL. If the resource content at that URL ever changes, a new and different ETag is assigned. Used in this manner ETags are similar to fingerprints, and they can be quickly compared to determine whether two versions of a resource are the same. Comparing ETags only makes sense with respect to one URL—ETags for resources obtained from different URLs may or may not be equal, so no meaning can be inferred from their comparison.

**Contents**

1 Deployment risks
2 Strong and weak validation
3 Typical usage
4 Tracking using ETags
5 References
6 External links

## Deployment risks [edit]

The use of ETags in the HTTP header is optional (not mandatory as with some other fields of the HTTP 1.1 header). The method by which ETags are generated has never been specified in the HTTP specification.

Common methods of ETag generation include using a collision-resistant hash function of the resource's content, a hash of the last modification timestamp, or even just a revision number.

In order to avoid the use of stale cache data, methods used to generate ETags should guarantee (as much as is practical) that each ETag is unique. However, an ETag-generation function could be judged to be "usable" if it can be proven (mathematically) that duplication of ETags would be "acceptably rare", even if it could or would occur.

Some earlier checksum functions, such as CRC32 and CRC64, are known to suffer from this hash collision problem. Because of this they are not good candidates for use in ETag generation.

## Strong and weak validation [edit]

The ETag mechanism supports both *strong validation* and *weak validation*. They are distinguished by the presence of an initial "W/" in the ETag identifier, as:

```
"123456789"   -- A strong ETag validator
W/"123456789"  -- A weak ETag validator
```

A strongly validating ETag match indicates that the content of the two resources is byte-for-byte identical and that all other entity fields (such as Content-Language) are also unchanged. Strong ETags permit the caching and reassembly of partial responses, as with byte-range requests.

A weakly validating ETag match only indicates that the two resources are semantically equivalent, meaning that for practical purposes they are interchangeable and that cached copies can be used. However the resources are not necessarily byte-for-byte identical, and thus weak ETags are not suitable for byte-range requests. Weak ETags may be useful for cases in which strong ETags are impractical for a web server to generate, such as with dynamically-generated content.

## Typical usage [edit]

In typical usage, when a URL is retrieved the web server will return the resource along with its corresponding ETag value, which is placed in an HTTP "ETag" field:

```
ETag: "686897696a7c876b7e"
```

The client may then decide to cache the resource, along with its ETag. Later, if the client wants to retrieve the same URL again, it will send its previously saved copy of the ETag along with the request in a "If-None-Match" field.

```
If-None-Match: "686897696a7c876b7e"
```

On this subsequent request, the server may now compare the client's ETag with the ETag for the current version of the resource. If the ETag values match, meaning that the resource has not changed, then the server may send back a very short response with an HTTP **304 Not Modified** status. The 304 status tells the client that its cached version is still good and that it should use that.

However, if the ETag values do not match, meaning the resource has likely changed, then a full response including the resource's content is returned, just as if ETags were not being used. In this case the client may decide to replace its previously cached version with the newly returned resource and the new ETag.

ETag values can be used in web page monitoring systems. Efficient web page monitoring is hindered by the fact that most websites do not set the ETag headers for web pages. When a web monitor has no hints whether web content has been changed all content has to be retrieved, and analyzed, using computing resources for both the publisher and subscriber.

## Tracking using ETags [edit]

ETags can be used to track unique users,[2] as HTTP cookies are increasingly deleted by privacy-aware users. In July 2011, Ashkan Soltani and a team of researchers at UC Berkeley reported that a number of websites, including Hulu.com, were using ETags for tracking purposes.[3] Hulu and KISSmetrics have both ceased "respawning" as of 29 July 2011,[4] as KISSmetrics and over 20 of its clients are facing a class-action lawsuit over the use of "undeletable" tracking cookies partially involving the use of ETags.[5]

Because ETags are cached by the browser, and returned with subsequent requests for the same resource, a tracking server can simply repeat any ETag received from the browser to ensure an assigned ETag persists indefinitely (in a similar way to persistent cookies). Additional caching headers can also enhance the preservation of ETag data.[6]

ETags may be flushable by clearing the browser cache (implementations vary).

## References [edit]


1. ^ "Editing the Web - Detecting the Lost Update Problem Using Unreserved Checkout" . *W3C Note*. 10 May 1999.
2. ^ "tracking without cookies" . 17 February 2003.
3. ^ "Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning" . 29 July 2011.
4. ^ "Respawn Redux" . 11 August 2011.
5. ^ AOL, Spotify, GigaOm, Etsy, KISSmetrics sued over undeletable tracking cookies
6. ^ Cookieless cookies (using ETags as cookies)

- ETag in HTTP/1.1 specification
- Concerning Etags and Datestamps by Lars R. Clausen (2004)


## External links [edit]

- Apache HTTP Server Documentation - FileETag Directive
- *Editing the Web: Detecting the Lost Update Problem Using Unreserved Checkout* , W3C Note, 10 May 1999.
- Old SQUID Development projects - ETag support (completed in 2001)
- Using ETags to Reduce Bandwidth & Workload with Spring & Hibernate
- Live demo of zombie cookie using ETags

Categories: Hypertext Transfer Protocol headers | Internet privacy | Cache (computing) | Proxy servers

This page was last modified on 13 February 2014 at 04:17.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view